الفصل الاول تمثيل البيانات

- عندما يتم ادخال البيانات بواسطة وحدات الادخال فانها تخزن داخل الذاكرة الداخلية
- وتتكون ذاكرة الحاسب من ملايين الخلايا (الاماكن) المتساوية وتسمى هذه الخلايا بالبايت (byte) وتحمل هذه الخلايا ارقاما متسلسلة تبدأ من الصفر
- تمثل عدد هذه الخلايا حجم ذاكرة الحاسب
- اى ان كلما زاد عدد هذه الخلايا (byte) كانت ذاكرة الجهاز اكبر وبالتالى فان سعة تخزين البيانات تكون اكبر
- تقاس ذاكرة بالكيلوبايت

انواع البيانات

نجد عند تخزين بيان ما فى الذاكرة نجد انه يشغل مساحة حجمها خلية واحدة (واحد بايت) وبيان اخر يشغل مساحة حجمها ٣ خلايا (٣ بايت) نجد هذا الاختلاف فى المساحة المحجوزة للبيان يرجع الى نوع البيان المخزن داخل هذه الخلايا

ومن الممكن ان يكون البيان المخزن داخل هذه الخلايا ثابت او متغير

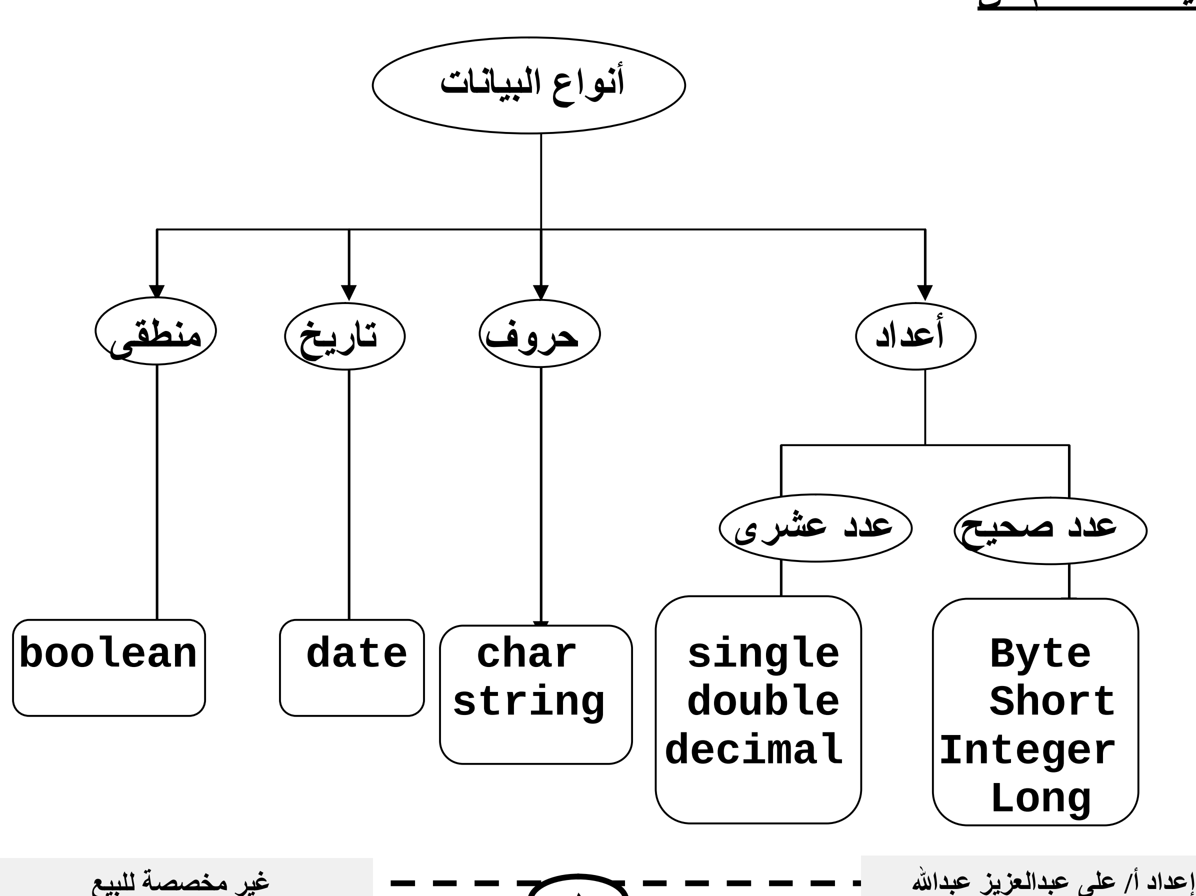
ثابت : هوالبيان الذى لاتتغير قيمته اثناء تشغيل البرنامج

ويطلق على إعطاء اسماء للخلايا الثابتة (التى تحمل قيم ثابتة لا تتغير) الإعلان عن الثوابت

متغير : هو البيان الذى تتغير قيمته اثناء تشغيل البرنامج

ويطلق على إعطاء اسم للخلايا المتغبرة (التى تحمل قيما متغيرة) الإعلان عن المتغيرات

انواع البيانات تنقسم الى



الجدول التالى يوضح نوع البيان وحجمه الذى يشغله فى الذاكرة وطبيعة البيان المخزن داخله ودرجة الدقة

م	نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت	طبيعة وحدود البيان المخزن داخله (المدى)	درجة الدقة (عدد الخانات)
١	رقم صحيح Byte	١	من ٠ الى ٢٥٥	لا توجد
٢	رقم صحيح Short	٢	من – ٣٢٧٦٨ الى ٣٢٧٦٧	٥ ارقام
٣	رقم صحيح Integer	٤	من -٢١٤٧٤٨٣٦٤٨ الى ٢١٤٧٤٨٣٦٤٧	١٠ ارقام
٤	رقم صحيح Long	٨	اكثر من ٢١٤٧٤٨٣٦٤٧	١٩ رقم
٥	Single عدد عشرى	٤	من $١,٤×١٠^{-٤٥}$ إلى $١,٤×١٠^{+٣٨}$	٧ ارقام
٦	Double عدد عشرى	٨	من $٤,٩×١٠^{-٣٢٤}$ إلى $١,٨×١٠^{+٣٠٨}$	١٥ أرقام
٧	decimal عدد عشرى	١٦	مبلغ من المال	٢٩ رقم
٨	Char حرفى	٢	تخزين حرف واحد	لا يوجد
٩	String حرفى	العديد من الخلايا	تخزين اعداد من ٠ الى ٢ بليون حرف	لا يوجد
١٠	Date تاريخ	٨	تاريخ من ١/١/٠٠٠١ الى ٣١/١٢/٩٩٩٩	لا يوجد
١١	Boolean منطقى	٢	True or False	لا يوجد

مثال توضيحى

لو اردنا تخزين العدد الصحيح ٢٠٠ فاننا يمكن تخزينه فى كل من byte ,short , integer , long ولكن الاصح تخزينه من نوع byte لانه بذلك سيشغل مساحة اقل فى الذاكرة

مثال آخر

لو اردنا تخزين العدد ٣٢١٤٥٦,١٥٤٧٨٩٦١ ٤١٧ داخل ثلاث متغيرات من نوع decimal , single , double

نجدان القيمة التى تخزن داخل متغير من نوع **single** هى : ٣٢١٤٥٦,٢

والقيمة التى تخزن فى متغير من نوع **double** هى ٣٢١٤٥٦,١٥٤٧٨٩٦١٤

والقيمة التى تخزن فى متغير من نوع **decimal** هى ٣٢١٤٥٦,١٥٤٧٨٩٦١٤١٧

الثوابت والمتغيرات

الثوابت : هى طريقة لاعطاء اسماء للقيم التى لا تتغير اثناء تشغيل البرنامج
ويوجد فى البرنامج نوعين من الثوابت
أ – ثوابت جوهرية: وهى ثوابت موجودة فى بناء وجوهر نظام vb.net اى انها ثوابت سابقة التعريف يستطيع اى برنامج vb.net ان يستعملها مباشرة بدون الاعلان عنها مثال للثوابت الجوهرية ثوابت الالوان Color Constant
ب- الثوابت المسماه : هى ثوابت يقوم المبرمج بتعريفها طبقا لاحتياجات البرنامج معنى ذلك انها غيرسابقة التعريف بالنسبة للبرنامج فيجب الاعلان عنها قبل استعمالها فى البرنامج والا سيحدث خطا
ويسمى هذا الخطأ syntax error معناه خطأ فى قواعد البرمجة
****

الصيغة العامة للاعلان عن الثوابت فى البرنامج

قيمة الثابت = نوع البيان الذى سيخزن فى الثابت as اسم الثابت Const

يجب اعطاء قيمة للثابت كما هو موضح بأعلى فى التعريف
اهمية الاعلان عن الثوابت بدلا من استخدام قيمها مباشرة
١- السهولة فى تغير قيمة الثابت اذا اردنا ذلك
٢- يصبح البرنامج مفهوما اكثر عند قرائته
****

المتغيرات :
هى القيم التى تتغير اثناء تشغيل البرنامج ويجب الاعلان عنها داخل البرنامج
الصيغة العامة للاعلان عن المتغيرات

نوع البيان الذى سيخزن فى المتغير as اسم المتغير Dim
****

قواعد تكوين الاسماء فى الفيجول بيزك vb.net

عند كتابة اسم المتغير او الثابت يجب اتباع الاتى حتى لايحدث خطأ (syntax error)
١- يجب ان يبدأ الاسم بحرف من الحروف الانجليزية سواء capital او small
٢- بعد الحرف الاول يمكن ان يأتى اى عدد من الحروف او الارقام او الرمز _ underscore باى ترتيب
٣- لا يسمح ان يحتوى الاسم على اى رموز او علامات خاصة كالمسافة والنقطة وعلامة الاستفهام وعلامة التعجب ....الخ
٤- لا يسمح ان يكون الاسم من الكلمات المحجوزة فى vb.net مثل (double,date,dim,const)

ملاحظات
- من المستحب ان يكون اسم المتغير او الثابت يدل على محتواه
- اذا اردنا ان يكون اسم المتغير من مقطعين والمسافة غير مسموحة فى اسم المتغير فالحل نبدأ كل مقطع بكابتل

مثال : first name خطأ لوجود مسافة فى اسم المتغير
انما FristName صحيحة كاسم للمتغير ؛ وايضا frist_name صحيحة لان لايوجد مسافات والرمز المستخدم هو underscore
- يفضل بعض المبرمجين ان يبدأ اسم المتغير بثلاثة احرف تعطى انطباعا بنوع المتغير مثال لو متغير اسمه frist وسيخزن به بيان من نوع integer فمن المستحسن ان يكون اسم المتغير Intfrist

اختيار النوع المناسب لكل متغير

يتحدد تلقائيا ثلاث خصائص لنوع المتغير فى اى برنامج عند اختيارنا لهذا المتغير وهذه الخصائص هى :

١- نوع البيانات التى يمكن تخزينها فيه
٢- العمليات التى يمكن اجراؤها عليه
٣- حجم الذاكرة الذى يحتاجه هذا المتغير

ما الفرق بين هذين التعريفين

```
Dim a as integer
Dim b as integer=٢٠
```

اولا كلاهما تعريف لمتغير لانه بدأ بكلمة **dim** ولكن الاختلاف ان قيمة المتغير **a** الابتدائية هى صفر وقيمة المتغير **b** الابتدائية هى ٢٠

ما الفرق بين هذين التعريفين

```
Dim a as string
Dim b as string= "sama"
```

كما شرحنا انهم تعريف لمتغير ولكن القيمة الابتدائية للمتغير **a** هى سلسلة حرفية فارغة "" والقيمة الابتدائية للمتغير **b** هى "sama"

****

هل السلسلة الحرفية "" تساوى السلسلة الحرفية "   " ؟

الاجابة لا

لان "" عبارة عن سلسلة حرفية فارغة

أما "   " فهى سلسلة حرفية بها مسافات(طولها هو عدد المسافات)

****

اعلان عن مجموعة من المتغيرات باستعمال أمر Dim واحد

مثال توضيحى

اكتب صيغة تعريف المتغيرات الاتية

المتغيرين **a,b** نوع البيان الذى سيخزن بداخلهم **integer**

المتغيرين **c,d** نوع البيان الذى سيخزن بداخلهم **string**

١-

```
Dim a as integer
Dim b as integer
Dim c as string
Dim d as string
```

٢-

```
Dim a,b as integer
Dim c,d as string
```

٣-

```
Dim a,b as integer , c,d as string
```

الثلاث صور السابقة لتعريف المتغيرات كلها صحيحة ولكن نلاحظ فى الطريقة الاخيرة اننا اعلنا عن جميع المتغيرات باستخدم امر ..... واحد

****

امر التخصيص (الاحلال)

ان اهم دور للمتغيرات فى البرنامج هو تخزين البيانات بها ولعمل ذلك نستخدم امرالتخصيص وهذا الامر يأخذ الشكل الاتى

القيمة = اسم المتغير

وعلامة= تسمى معامل التخصيص
يجب ان تكون القيمة هى التى على يمين = ، واسم المتغير على شمال =

A = ١٠

١٠ = A

الصيغة الاولى صحيحة كأمر تخصيص لان معناها ان القيمة ١٠ سيتم تخزينها داخل المتغير A
اما الثانية الصيغة خطأ لان معناها ان قيمة المتغير A سيتم تخزينها فى ١٠ نفهم من ذلك ان القيمة الموجودة على يمين = هى التى ستخزن فى المتغير الموجود على يسار =
يجب ان تكون القيمة المراد تخزينها من نوع ملائم لنوع المتغير الذى سيتم التخزين به
مثال :

```
Dim a as integer
a ="mohamed"
```

نجد فى المثال السابق اننا حجزنا مكان فى الذاكرة لمتغير اسمه a و سيتم تخزين بيانات عددية صحيحة بداخله ولكن هنا نلاحظ فى الخطوة الثانية انه تم ادخال بيان حرفى وهذا خطأ ولكن الصح مثلا a=٧ عدد

القيمةالتى تخزن فى المتغير اما ان تكون

١- قيمة مجردة مثال A=٤

****

٢- قيمة متغير اخر مثـــال

١- a = ٢ , b = ٥
٢- a = b

م	بيان	قيمة b	قيمة a
١	قبل عملية التخصيص (الاحلال)	٥	٢
٢	بعد عملية التخصيص(الاحلال)	٥	٥

٣- ناتج عملية حسابية

١- a = ٤ , b = ٧
٢- a = b +٢

م	بيان	قيمة b	قيمة a
١	قبل عملية التخصيص (الاحلال)	٧	٤
٢	بعد عملية التخصيص(الاحلال)	٧	٩

نلاحظ فى ٢ ،٣ ان قيمة المتغيرات التى على يمين = لا تتغير وانما قيمة المتغيرات الموجودة على شمال = هى التى تتغير
وايضا ان القيمة الجديدة للمتغير تحل محل القيمة القديمة
مفهوم العلامة = فى امر التخصيص يختلف عن مفهومها فى المعادلات الرياضية
مثال :٣ = X + ١
تمثل معادلة رياضية ولكن غير صحيحة كأمر تخصيص لوجود عملية حسابية على يسار =
X = x + ١
لا تمثل معادلة رياضية ولكنها صحيحة كأمر تخصيص ومعناها ان قيمة المتغير الجديدة

لـــ x تساوى قيمة المتغير القديمة + ١

كيفية تنفيذ امر التخصيص

١- تحديد القيمة الموجودة على يمين = كالاتى
أ- استبدال كل متغير او ثابت بقيمته
ب- اجراء العملية الحسابية واستخراج النتيجة
٢- تخزين النتيجة فى المتغير على شمال =
يعنى ذلك أنه فى عملية التخصيص يتم احتساب القيم الموجودة على يمين علامة تساوى أولا ثم تخزينها فى المتغير الموجود على يسار معامل التخصيص

****

استعمال امر التخصيص مع المتغيرات الحرفية

كما اننا نقوم بتخزين ناتج العمليات الحسابية فى المتغيرات الرقمية باستخدام امر التخصيص فاننا نستطيع ايضا تخزين القيم الحرفية فى متغيرات حرفية باستعمال نفس الامر

وايضا كما اننا نقوم باجراء عمليات رياضية باستعمال المتغيرات الرقمية فاننا نستطيع بالمثل اجراء بعض العمليات على المتغيرات الحرفية
من اشهر العمليات التى يمكن ان نقوم بها على المتغيرات الحرفية عملية ربط اكثر من قيمة حرفية او متغير حرفى ببعضها لتكوين متغير حرفى واحد

هذا الربط يتم باستعمال الرمز & الذى يسمى معامل الربط

مـثـال :

```
"number" &"  " & "of student" = "number of student"
```

****

التحويل بين الانواع المختلفة للبيانات

تختلف طريقة تخزين الانواع المختلفة من البيانات فى ذاكرة الحاسب كما يختلف ايضا حجم الذاكرة (عدد الخلايا) التى يتم التخزين بها . فمثلا تختلف طريقة تخزين القيمة الحرفية ١٢٣ اختلافا تاما عن طريقة تخزين القيمة الرقمية ١٢٣ فمثلا يتم تخزين القيمة الحرفية ١٢٣ فى ثلاث خلايا اما القيمة الرقمية ١٢٣ فمن الممكن تخزينها فى خلية واحدة

١- تحويل البيانات الحرفية الى بيانات رقمية

عند ادخال البيانات اثناء تشغيل البرنامج فاننا ندخلها عن طريق اداة الكتابة **TextBox** تكون هذه البيانات هى محتوى الخاصية(**text**)
ولكن الفيجول بيزيك يتعامل مع محتوى الخاصية **text** على انها بيان حرفى وليس عددي فى هذه الحالة لايمكن اجراء عمليات حسابية على هذه البيانات مباشرة
ولذلك اذا اردنا اجراء العمليات الحسابية على البيانات المدخلة بواسطة اداة **TextBox**فيجب تحويلها اولا الى قيم عددية ويتم ذلك من خلال استخدام الوسيلة **Parse**

الشكل العام لدالة Parse

لتحويل قيمة حرفية الى رقم عدد صحيح

Integer.parse(القيمة الحرفية التى تريد تحويلها)

لتحويل قيمة حرفية الى رقم عددعشرى

Decimal.parse(القيمة الحرفية التى تريد تحويلها)

وهكذا يتم كتابة نوع المتغير الرقمى المناسب ثم Parse ثم القيمة الحرفية المطلوب تحويلها

****

٢- تحويل البيانات الرقمية الى بيانات الحرفية

كما سبق وشرحنا انه يجب تحويل القيمة المدخلة بواسطة اداة TextBox الى قيم عددية لاجراء العمليات الحسابية عليها فاننا ايضا لكى نظهر هذه البيانات فى أداة TextBox أو أداة Label يجب تحويلها الى قيم حرفية ولعمل ذلك نستخدم الدالة Tostring

الشكل العام للدالة Tostring

(القيمة العددية المراد تحويلها).Tostring

****

مجموعة اوامرTry/Catch

عندما يتم تحويل البيان المدخل بواسطة اداة TextBox الى بيان رقمى بالطريقة السابقة وعندما يقوم مستخدم البرنامج بادخال قيمة حرفية ينتج عن ذلك توقف البرنامج وظهور رسالة تفيد حدوث خطأ ومعظم مستخدمى البرامج لا يستطيعون التعامل مع هذه الرسائل لعدم فهمها

فلذلك يفضل المبرمجين استعمال الامر try/catch لان هذا يتيح لهم التعامل مع الاخطاء بانفسهم وعرض رسائل أوضح لمستعمل البرنامج يقوم المبرمجون بكتابتها

ولذلك فان مجموعة وامر try/catch تستخدم لاكتشاف الاخطاء التى تحدث اثناء تشغيل البرنامج والتعامل معها

****

الصيغة العامة لاستخدام اوامرtry/catch

```
  Try
الجملة المحتمل مستخدم البرنامج ان يخطئ فيها
Catch
الرسالة التى تظهر لمستخدم البرنامج موضح بها الاخطاء التى حدثت
End try
```

ابسط شكل للرسالة Msgbox("الرسالة")

****

العمليات الحسابية

الجدول الاتى يوضح العمليات الحسابية التى يمكن اجراؤها فى vb.net

م	العملية	المعامل	مثال	الناتج
١	جمع	+	٢٣+٨	٣١
٢	طرح	-	٤٠-٧	٣٣
٣	ضرب	*	٦*٥	٣٠
٤	قسمة عادية	/	٣٠/٤	٧,٥
٥	قسمة اعداد صحيحة	\	٣١\٧	٤
٦	باقى القسمة	mod	٢٠mod٣	٢
٧	الاسس	^	٣^٤	٨١

قسمة الاعداد الصحيحة وباقى القسمة

قسمة الاعداد الصحيحة هى قسمة عدد صحيح على عدد صحيح واهمال باقى القسمة لتكون النتيجة عددا صحيحا

مثال

قسمة عادية ٤٢/٤= ١٠,٢٥

قسمة أعداد صحيحة ٤٢\٤=١٠

باقى القسمة ٤٢ mod ٤ = ٢

الاسس ^

يستعمل معامل الاس فى رفع عدد معين (الاساس) الى اس معين وتكون التيجة من النوع double

اولويات تنفيذ العمليات الحسابية

جميع لغات البرمجة بما فيها vb.net قد وضعت قاعدة تسمى قاعدة اولويات تنفيذ العمليات الحسابية لتحديد اولوية تنفيذ العملية الحسابية وفى هذه القاعدة يتم ترتيب المعاملات المختلفة ترتيبا تنازليا حسب اولوية تنفيذها عند اجتماعها فى عملية حسابية واحدة

ترتيب العمليات الحسابية فى vb.net

١- العمليات التى بين الاقواس ٢- عمليات الاسس ٣- عمليات الضرب والقسمة

٤- عملية قسم الاعداد الصحيحة ٥- عمليات حساب باقى القسمة ٦- عمليات الجمع والطرح

فى حالة وجود اكثر من عملية لها نفس الاولوية كالضرب والقسمة او والجمع والطرح فان ترتيب تنفيذها يكون من اليسار الى اليمين ، اما اذا تداخلت الاقواس (كأن يكون هناك اقواس داخل اقواس) فان الاقواس الداخلية يتم تنفيذ ما بينها اولا

****

ما نتيجة تنفيذ العليات الاتية:

(ب) ٢+١*٣^٢\٤	(أ)٣+٢*٢^٢/٤-١+٥*٣
٢+١*٩\٤	٣+٢*٤/٤-١+٥*٣
٢+٩\٤	٣+٨/٤-١+١٥
٢+٢	٣+٢-١+١٥
٤	٤+١٥
	١٩

قم بتتبع مجموعة الاوامر التالية:

١) dim x as short
٢) dim y as short
٣) dim z as short
٤) x=١٠
٥) x=٢*x+١
٦) y=x-٨
٧) z=y/٤
٨) z=z+٤

الحل من تعريف المتغيرات نجد ان نوع البيان الذى سيخزن بها هو عدد صحيح كما ان القيمة الابتدائية لكل منهم تساوى الصفر

الخطوة ورقمها		X	Y	Z
١	dim x as short	٠	Don't exit	Don't exit
٢	dim y as short	٠	٠	Don't exit
٣	dim z as short	٠	٠	٠
٤	x=١٠	١٠	٠	٠
٨	x=٢*x+١	٢١	٠	٠
٦	y=x-٨	٢١	١٣	٠
٧	z=y/٤	٢١	١٣	٣
٨	z=z+٤	٢١	١٣	٧

كلمة Don't exit تشير إلى أن المتغير لم يتم الإعلان عنه وفى الخطوة رقم ٧ عند قسمة قيمة المتغير(١٣) على ٤ فالناتج يكون ٣,٣٣٣٣٣٣ ونظرا لأن المتغير عددى صحيح فيتم التقريب لأقرب رقم صحيح وهو ٣

لتصميم اى مشروع فى vb.net

اولا تصميم واجهة البرنامج (Form)

١ - تحديد الأدوات المطلوبة ورسمها أو وضعها على النموذج Form

٢ - تنسيق الأدوات من خلال صندوق الخصائص (الخلفية – نوع الخط – حجم الخط – ٠٠٠٠٠٠٠٠٠٠)

ثانيا كتابة الكود

١ - تحديد الاداة التى سيتم كتابة الكود بداخلها(الاداة التى سيتم الضغط عليها لتنفيذ الامر) ثم النقر المزدوج عليها للانتقال إلى نافذة البرمجة وكتابة الكود المناسب (معالج الأحداث)

٢ - داخل كود البرمجة:

أ- اولا تعريف المتغيرات او الثوابت الموجودة بالبرنامج

ب- تخزين البيانات داخل المتغيرات باستخدام امر التخصيص

جـ- اجراء العمليات المختلفة على البيانات

د- يمكن كتابة أوامر لتغيير خصائص بعض الكائنات بالأمر

Objectname.property=value

هـ- كتابة بعض الوسائل لبعض الأدوات بالأمر

Objectname.method

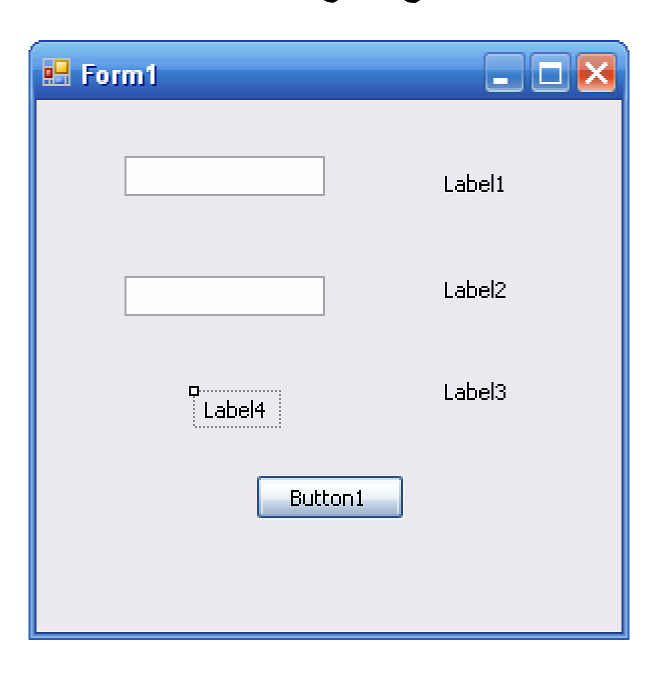
ثالثا تنفيذ البرنامج:

١ - النقر على مفتاح F٥ من لوحة المفاتيح أو من قائمة Debug نختار Debug Start أو من شريط الأدوات ننقر على الأداة ▶ لتشغيل البرنامج

٢ - القيام بالحدث المناسب لتنفيذ معالج الأحداث (النقر على الأداة التى تم كتابة الكود لها) وذلك لتنفيذ الكود (الأوامر) التى تم كتابتها وعرض النتائج المطلوبة

****

اكتب مشروع لجمع عددين






```
Private Sub Button١_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button١.Click
    Dim first, second, result As Decimal
     first = Decimal.Parse(TextBox١.Text)
     second = Decimal.Parse(TextBox٢.Text)
     result = first + second
     Label٤.Text = result.ToString
  End Sub
```

نجد عند تنفيذ هذا الكود عند ادخال قيمة حرفية يتوقف البرنامج ويعرض رسالة خطأ

البرنامج باستعمال الامر try

```
Private Sub Button١_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button١.Click
    Dim first, second, result As Decimal
    try
    first = Decimal.Parse(TextBox١.Text)
    second = Decimal.Parse(TextBox٢.Text)
    catch
    msgbox("لقد قمت بادخال قيمة حرفية")
    End try
    result = first + second
    Label٤.Text = result.ToString
End Sub
```



التحويل بين أنواع البيانات العددية

قد نحتاج فى بعض الأحيان إلى إجراء عمليات تحويل على قيم عددية وتحويلها إلى قيم عددية أخرى بمعنى إننا نحتاج إلى تحويل قيمة عددية ذات سعة تخزينية صغيرة إلى قيمة عددية ذات سعة تخزينية أكبر والعكس ومن هنا يأتى دور

١ ـ التحويل المباشر ٢ ـ التحويل الغير مباشر

أولاً ـ التحويل المباشر

وهى عملية تحويل قيمة عددية ذات سعة تخزينية صغيرة (عدد الخلايا) إلى قيمة عددية ذات سعة تخزينية أكبر . بمعنى وضع محتوى صغير فى محتوى أكبر منه . وذلك من خلال

النوع المحول	عدد الخلايا	النوع المحول إليه
Byte	١	Short , Integer , Long , Decimal , Single , Double
Short	٢	Integer , Long , Decimal , Single , Double
Integer	٤	Long , Decimal , Single , Double
Long	٨	Decimal , Single , Double
Decimal	١٦	Single , Double
Single	٨	Double

وبالتالى يتم تحويل القيم العددية الأصغر إلى قيم عددية أكبر وتسمى (التحويل المباشـــر)

مثال : تحويل القيمة العددية من النوع (Integer) إلى النوع (Decimal)

```
Dim  X  As  Integer
X  =  Decimal . Parse(x)
```

ثانياً – التحويل الغير مباشر

وهى عملية تحويل قيمة عددية ذات سعة تخزينية كبيرة (عدد الخلايا) إلى قيمة عددية ذات سعة تخزينية أقل . وذلك خلال التصنيف (Convert) وهو أمر خاص بالتحويل الغير مباشر .

مثال : تحويل القيمة العددية من النوع (Decimal) إلى النوع (Integer)

```
Dim  X  As  Decimal
Dim  Y  As  Integer
Y  =  Convert . ToInteger (X)
```

والجدول التالى يوضح تحويل قيمة

المتغير (X) من النوع (Decimal) إلى المتغير العددى (Y) من النوع (Integer)

كيفية إجراء عملية التحويل الغير مباشر

النوع	الكود
Short	ToInt١٦
Integer	ToInt٣٢
Long	ToInt٦٤
Decimal	ToDecimal
Single	ToSingle
Double	ToDouble

ملحوظة :

عند استخدام التحويل غير المباشر يجب ملاحظة ومراعاة الآتى:

١ـ ألا تؤثر عملية التحويل على الخانات المؤثرة أى خانات الرقم الصحيح.

٢ـ ألا تستخدم عملية التحويل غير المباشر إلا فى حالة التأكد من أن النوع المحول إليه يستوعب القيمة التى يتم تحويلها.

٣ـ يتم تقريب الأرقام العشرية فى حالة التحويل إلى أرقام صحيحة.

الفصل الثانى الشروط والقرارات

فى كثير من الأحيان عند كتابة أى برنامج فإننا نحتاج إلى شرط معين لتنفيذ هذا البرنامج وبناء عليه لابد من اتخاذ قرار معين لذلك نلاحظ أن اتخاذ أى قرار لابد وأن يكون مرتبط بشرط معين .

التعبير الشرطى : هو عبارة عن صياغة للعلاقة بين كميتين . وللتعبير الشرطى قيمة تتوقف على صحة العلاقة أو عدم صحتها.

فإذا كانت العلاقة صحيحة فتكون قيمة التعبير (True) .

أما إذا كانت العلاقة غير صحيحة تكون قيمة التعبير (False)

ولصياغة التعبيرات الشرطية نقوم باستعمال بعض معاملات المقارنة التالية والتى يوضحها المثال التالى .

مثال : إذا كانت قيمة المتغير (A = ٥٠) ، قيمة المتغير (B = ٦٠) فإن العلاقة هى :-

الناتج	العملية	العلاقة	المعامل
False	A > B	Greater than أكبر من	>
True	A < B	Less than أصغر من	<
False	A = B	Equal to يساوى	=
True	A< > B	Not equal to لا يساوى	< >
False	A >= B	Greater than or equal أكبر من ويساوى	>=
True	A <= B	Less than or equal أصغر من ويساوى	<=

١- جملة (الأمر) IF--------------Then -----------------Else

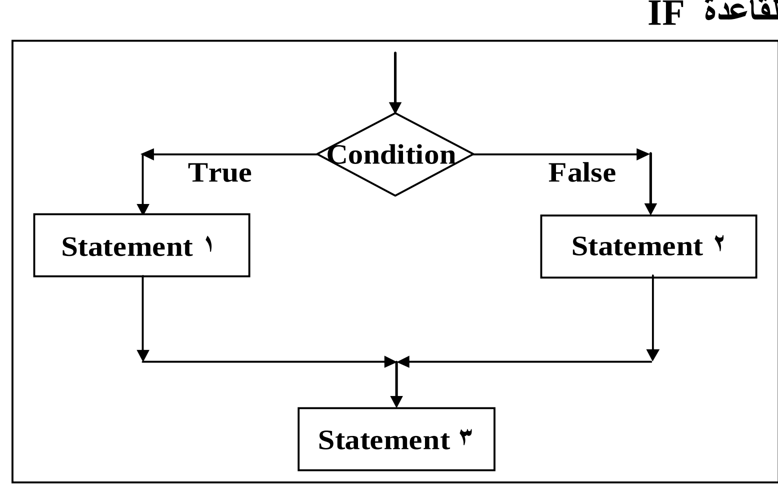
وهى تسمى الجملة الشرطية وأن الأمر (IF .. Then... Else) هو أمر يساعد فى الإجابة على سؤال معين بناء على قيمة هذا التعبير الشرطى. بمعنى

فإذا كان نتيجة الشرط (True / Yes) سوف يتم تنفيذ مجموعة من الأوامر.

أما إذا كانت نتيجة الشرط (False / No) سوف يتم تنفيذ مجموعة أخرى من الأوامر.

الشكل العام لقاعدة IF

```
IF Condition Then
      Statement ١
Else
      Statement ٢
End IF
```



مثال : اكتب برنامج لحساب الراتب الأسبوعى لموظف بشرط إذا كانت ساعات عمل الموظف أقل من أو تساوى ٤٠ ساعة فإن أجر الساعة يكون ٥ جنيهات أما إذا كانت ساعات العمل أكثر من ٤٠ ساعة تحسب أجر الساعة ١٠ جنيهات عن كل ساعة تزيد عن ٤٠ ساعة.

فى هذا المثال يلاحظ أن أجر أو راتب الموظف عبارة عن فئتان الأولى أجر الساعة خمسة جنيهات إذا كانت ساعات عمله لا تتعدى ٤٠ ساعة أسبوعيا. والثانية أن ما يزيد عن ٤٠ ساعة يكون أجر الساعة الزائدة ١٠جنيهات بالإضافة للأجر العادى للساعات الـ ٤٠ الأولى وهى خمسة جنيهات وتتبع خطوات الحل التالية:

(أ) أعطاء أسماء للمتغيرات

المرتب Salary

ساعات العمل HW = Hours Work

الإضافى OT = Over Time

(ب) حساب الراتب الأسبوعى للموظف

١- إذا كانت ساعات العمل لا تزيد عن (٤٠ ساعة)

الراتب = عدد ساعات العمل * ٥ جنيهات

٢- إذا كانت ساعات العمل تزيد عن (٤٠ ساعة)

الأجر الإضافى = (عدد ساعات العمل – ٤٠ ساعة) * ١٠ جنيهات

الراتب = (٤٠ * ٥) + الأجر الإضافى

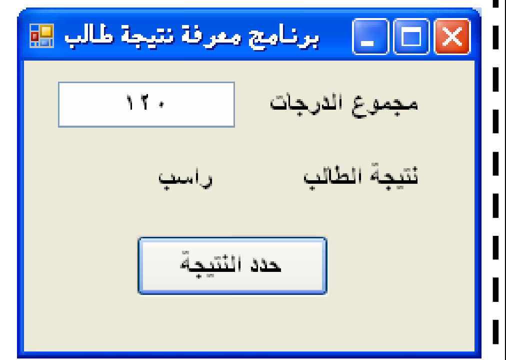
```
Dim HW, OT, Salary as Decimal
        HW = TextBox١.Text
IF HW > ٤٠ Then
        OT = (HW – ٤٠) * ١٠
        Salary = (٤٠ * ٥) + OT
Else
        Salary = HW * ٥
End IF
        Msgbox (“The Salary = “& salary)
```

حساب الراتب
50
الراتب
WindowsApplication1
The Salary = 300
OK

مثال٢: أكتب برنامج لمعرفة نتيجة (ناجح أم راسب) إذا علمت أن يعتبر الطالب ناجح إذا كان مجموع درجاته أكبر من أو يساوى ١٥٠ درجة ويكون راسب إذا كان مجموع درجاته أقل من ١٥٠ درجة.

```
Dim total As Decimal
    total = Me.TextBox١.Text
    If total >= ١٥٠ Then
      Me.Label٣.Text = "ناجح"
    Else
      Me.Label٣.Text = "راسب"
    End If
```

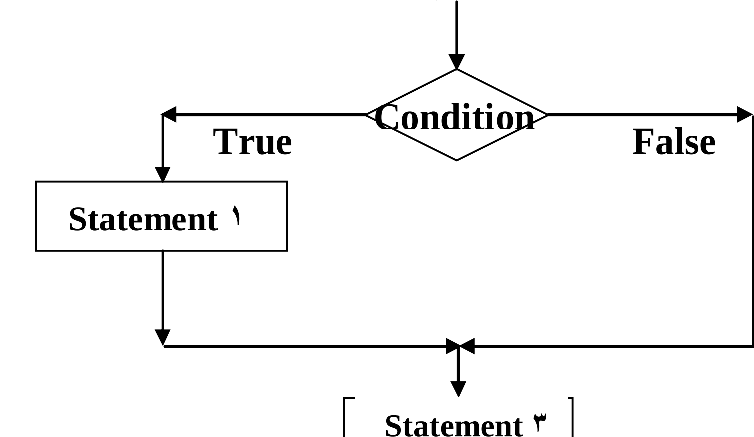
برنامج معرفة نتيجة طالب
مجموع الدرجات ١٧٥
نتيجة الطالب ناجح
حدد النتيجة



٢- جملة IF … Then

يمثل الأمر (IF .. Then) صورة مبسطة للأمر الأصلى (IF ... Then... Else) حيث أن الأمر (IF ... Then) يتم التعامل فيه مع شرط واحد فقط. بحيث عند تحقق الشرط يتم تنفيذ مجموعة من الأوامر وعند عدم تحقق الشرط لا يتم تنفيذ شيء .


والشكل العام لهذه الجملة البسيطة كما هو موضح




```
IF Condition Then
   Statement ١
End IF
     Statement ٣
```

<u>من الشكل يتضح أن:</u>

- **إذا كانت قيمة الشرط (True) سوف يتم تنفيذ مجموعة الأوامر الموجودة فى (Statement١) ثم تنفيذ مجموعة الأوامر الموجود فى (Statement ٣)**
- **إذا لم يتحقق الشرط (False) سوف يتم تنفيذ مجموعة الأوامر الموجودة فى (Statement ٣) مباشرة وتجاهل مجموعة الأوامر الموجودة فى (Statement ١)**

مثال: اكتب برنامج لحساب العمولة. عندما تصل المبيعات عن (١٠٠٠) جنية تصبح العمولة (١٠ %)

<u>Button١</u>

```
Dim X, Y as Decimal
    X = Textbox١.Text
IF X >= ١٠٠٠ Then
    Y = X * ٠,١٠
End IF
   Label١.Text = Y
```





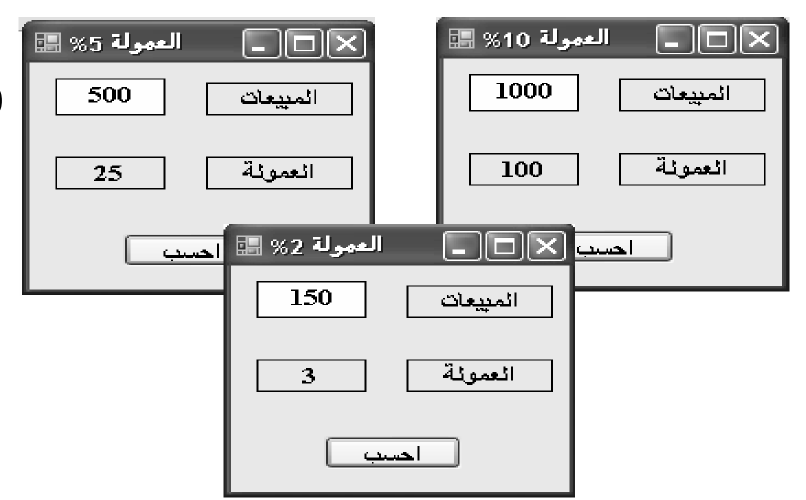
جملة IF … Then … Else IF

وهى تسمى بالشروط المتداخلة .. ويستخدم الشرط المتداخل فى حالة ما إذا كنا سنتخذ قراراً معيناً بناءً على تحقيق شرط معين إذا تحقق ينفذ أمر معين وإذا لم يتحقق يتم اختبار شرط ثانى وإذا تحقق الشرط الثانى ينفذ أمر آخر وإذا لم يتحقق يتم عمل اختبار شرطى جديد وهكذا وإذا لم تتحقق أى من هذه الشروط ينفذ الأمر الافتراضى والذى يأتى بعد كلمة Else ثم تنتهى الجملة كالعادة بكلمة End IF

مثال: اكتب برنامج لحساب العمولة عند مستويات مختلفة من المبيعات

Button١

```
Dim X, Y As Decimal
X = Decimal . Parse (TextBox١.Text)
 IF  X >= ١٠٠٠ Then
     Y = X * ٠,١
 ElseIf X >= ٥٠٠ Then
         Y = X * ٠,٠٥
 Else
         Y = X * ٠,٠٢
End If
        Label٣.Text = Y
```



مثال: اكتب برنامج لحساب تقديرات الطلاب عند مستوى معين من الدرجات إذا علمت أن المجموع الكلى للدرجات ١٠٠درجة وتكون تقديرات الطلبة : ممتاز أكبر من أو تساوى ٨٥درجة ، جيد جدا أكبر من أو تساوى ٧٥ ،جيد أكبر من أو تساوى ٦٥ ، مقبول أكبر من أو تساوى ٥٠ وكل تقدير حتى أقل من التقدير السابق لـه وخلاف ذلك يكون ضعيف (يشار للمجموع بالمتغير Total والتقدير بالمتغير Result) ثم عرض قيمة Result داخل أداة العنوان

Button١ برمجة الزر (احسب التقدير)

```
   Dim Total As Single, result As String
      Total = TextBox١.Text
      If Total >= ٨٥ Then
         result = " ممتاز "
      ElseIf Total >= ٧٥ Then
         result = " جيد جدا "
      ElseIf Total >= ٦٥ Then
         result = " جيد "
      ElseIf Total >= ٥٠ Then
         result = " مقبول "
      Else
         result = " ضعيف "
      End If
      Label٣.Text = result
```



مثال : اكتب برنامج باستخدام (VB.Net) لحساب المرتب من خلال عدد سنوات الخبرة وعدد ساعات العمل وذلك من خلال لجدول التالى :

المتغيرات المستخدمة	أجر الساعة (Wage Rate)	سنوات الخبرة (Service Years)
سنوات الخبرة ys (Year Service)	٥	< ٥
ساعات العمل hw (Hour Work)	٧	>=٥ And <١٠
أجر الساعة hr (Hour Rate)	١٠	>=١٠ And <١٥
المرتب salary	١٢	>=١٥ And <٢٠
	١٥	>=٢٠




```
Dim ys As Short
    Dim hw, hr, salary As Decimal
    Try
      ys = Short.Parse(TextBox١.Text)
      hw = Decimal.Parse(TextBox٢.Text)
      If ys >= ٢٠ Then
        hr = ١٥
      ElseIf ys >= ١٥ Then
        hr = ١٢
      ElseIf ys >= ١٠ Then
        hr = ١٠
      ElseIf ys >= ٥ Then
        hr = ٧
      Else
        hr = ٥
      End If
      salary = hr * hw
     Label٤.Text = salary.ToString
      MsgBox("أجر الساعة =  " & hr & vbCrLf & "عدد ساعات العمل = " & hw & vbCrLf & "
المرتب = " & salary, , "برنامج حساب مرتبات العاملين")
    Catch
      MsgBox("تأكد من سنوات الخبرة أو عدد ساعات العمل ", , " Salary ")
    End Try
  End Sub
```

ملحوظة:

١- الأمر (Vbcrlf) هو أمر لعمل تنسيق على المخرجات تظهر داخل صندوق الرسائل على سطر جديد وليس فى سطر واحد

٢- وجود (صندوقين) للرسائل الأول لإظهار ناتج المرتب.. والثانى لإظهار رسالة خطأ فى حالة إدخال بيانات خاطئة من خلال (Try / Catch)

الشروط المركبة

إذا كان أحد الشرطين أو كلاهما صحيحا يكون الشرط المركب صحيح	١- Or
إذا كان الشرط الأول صحيح يكون الشرط المركب صحيح ويكون الشرط الثانى لا قيمة له	٢- OrElse
إذا كان كل من الشرطين صحيح يكون الشرط المركب صحيح	٣- And

مقارنة القيم والمتغيرات الحرفية Strings

يتيح برنامج VB.Net إجراء مقارنة بين قيم حرفية وقيم حرفية أخرى (متغيرات حرفية – ثوابت حرفية خصائص حرفية – قيم حرفية مجردة)

وتتم عملية المقارنة بمقارنة الرمز الأول (الموجود فى أقصى يسار القيم الحرفية) للمتغير الأول بالرمز الأول المناظر للمتغير الثانى ثم نتجه رمزا ناحية اليمين وبمجرد وجود رمز مختلف تتوقف عملية المقارنة، وتكون النتيجة أن المتغير أو القيمة ذو الموجود بها الرمز ذو الرتبة الأقل هو الأصغر من الأخر

ويتم تحديد رتبة كل رمز طبقا لطريقة تخزين الرموز فى ذاكرة الكمبيوتر والتى تسمى ANSI Code حيث يتم إعطاء كل رمز (حرف – رقم – رمز خاص) رتبة معينة وهذه الرتبة هى رقم ثابت لا يتغير كما هو موضح فى الجدول التالى

الرمز	القيمة	الرمز	القيمة	الرمز	القيمة	الرمز	القيمة
h	١٠٤	P	٨٠	٨	٥٦	space	٣٢
i	١٠٥	Q	٨١	٩	٥٧	!	٣٣
j	١٠٦	R	٨٢	:	٥٨	"	٣٤
k	١٠٧	S	٨٣	;	٥٩	#	٣٥
l	١٠٨	T	٨٤	<	٦٠	$	٣٦
m	١٠٩	U	٨٥	=	٦١	%	٣٧
n	١١٠	V	٨٦	>	٦٢	&	٣٨
o	١١١	W	٨٧	?	٦٣	.	٣٩
p	١١٢	X	٨٨	@	٦٤	)	٤٠
q	١١٣	Y	٨٩	A	٦٥	(	٤١
r	١١٤	Z	٩٠	B	٦٦	*	٤٢
s	١١٥	[	٩١	C	٦٧	+	٤٣
t	١١٦	\	٩٢	D	٦٨	,	٤٤
u	١١٧	]	٩٣	E	٦٩	-	٤٥
v	١١٨	^	٩٤	F	٧٠	.	٤٦
w	١١٩	_	٩٥	G	٧١	/	٤٧
x	١٢٠	،	٩٦	H	٧٢	٠	٤٨
y	١٢١	a	٩٧	I	٧٣	١	٤٩
z	١٢٢	b	٩٨	J	٧٤	٢	٥٠
{	١٢٣	c	٩٩	K	٧٥	٣	٥١
|	١٢٤	d	١٠٠	L	٧٦	٤	٥٢
}	١٢٥	e	١٠١	M	٧٧	٥	٥٣
~	١٢٦	f	١٠٢	N	٧٨	٦	٥٤
Del	١٢٧	g	١٠٣	O	٧٩	٧	٥٥

حدد أى من الشروط التالية True وأيها False

الشرط	النتيجة	الشرط	النتيجة
"NEHAD"<"NEHAL"	العلاقة صحيحة نظرا لآن D أقل من L فى الرتبة	"١٢ Geeza" < "Cairo"	العلاقة صحيحة نظرا لآن الرقم أقل من الحرف فى الرتبة
"SALAH"<"SALAH ALI"	العلاقة صحيحة نظرا لآن Space أقل من أى حرف فى الرتبة	"samy" < "Samy"	العلاقة خطأ نظرا لآن s smallأكبر فى الرتبة

استخدام جملة (IF) مع أداتى

Checkbox, RadioButton

الأداة (RadioButton)

هى أداة تستخدم فى عرض مجموعة من البدائل المتاحة على أن يقوم المستخدم باختيار بديل واحد فقط

خصائص الأداة RadioButton :

الخاصيـــــــــة	الوظيفة
Name	تحديد اسم الأداة داخل نافذة البرمجة وعادة ما يبدأ الاسم بـ rad
Backcolor	تحديد لون خلفية الأداة
Appearance	تحديد شكل الأداة (تقليدى – أو Button)
Checked	تحديد إمكانية اختيار الأداة أم لا (True / False)
Enabled	تحديد إمكانية جعل الأداة متاحة أم لا
Forecolor	تحديد لون خط الكتابة على الأداة
Height	تحديد ارتفاع الأداة
Image	إمكانية وضع صورة على الأداة
Text	تغيير اسم الأداة أو النص الذى يظهر على الأداة
Visible	جعل الأداة (مرئية / مخفية)
Width	تحديد عرض الأداة

ملحوظة:

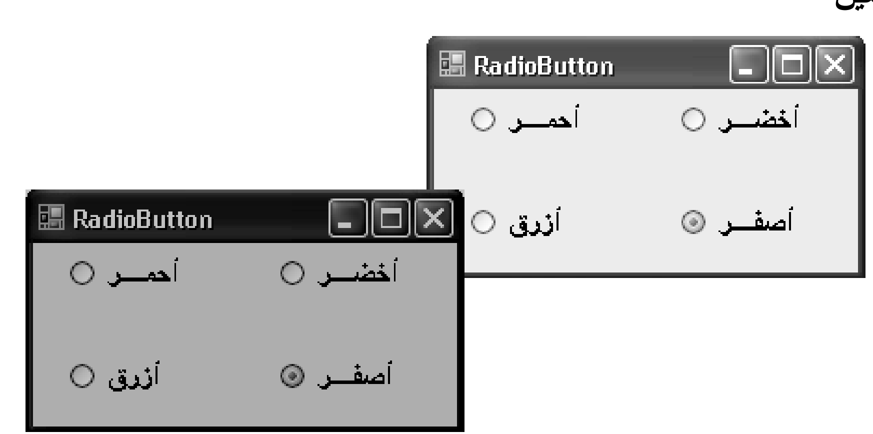
إذا كان لديك (٣ RadioButton) وأردت جعل واحدة منها نشطة. أى تم اختيار (RadioButton٢) يتم كتابة الأمر التالى RadioButton٢.Checked = True

سوف نلاحظ أن:

تم اختيار (RadioButton٢) وجعلها نشطة أى تصبح الخاصية Checked قيمتها (True) وتصبح باقى (RadioButton) الأخرى غير نشطة أى أن الخاصية Checked قيمتها (False)

وإذا تم جعل الخاصية checked=true لأداة أخرى تصبح هذه الخاصية لباقى الأدوات false

مثال : اكتب برنامج لتغيير لون الخلفية للنموذج فى حالة وجود (٤ RadioButton)بحيث تختص كل واحدة بلون معين



RadioButton١

```
Me.Backcolor = Color.Red
```

RadioButton٢

```
Me.Backcolor = Color.Green
```

RadioButton٣

```
Me.Backcolor = Color. Blue
```

RadioButton٤

```
Me.Backcolor = Color. Yellow
```

الأداة (CheckBox)

هى أداة تتيح عرض كثير من البدائل لاختيار أى عدد منها.

خصائص الأداة CheckBox:

الخاصيــــــــــــة	الوظيفة
Name	تحديد اسم الأداة داخل نافذة البرمجة وعادة ما يبدأ الاسم بـ chk
Backcolor	تحديد لون خلفية الأداة
Appearance	تحديد شكل الأداة (تقليدى – أو Button)
Checked	تحديد إمكانية اختيار الأداة أم لا (True / False)
Enabled	تحديد إمكانية جعل الأداة متاحة أم لا
Forecolor	تحديد لون خط الكتابة على الأداة
Height	تحديد ارتفاع الأداة
Image	إمكانية وضع صورة على الأداة
Text	تغيير اسم الأداة أو النص الذى يظهر على الأداة
Visible	جعل الأداة (مرئية / مخفية)
Width	تحديد عرض الأداة

ملحوظة: ويمكن أيضاً اختيار الأداة من عدمه كما تم سابقاً من خلال الأوامر التالية

الأمر	النتيجة
CheckBox١.Checked = True	جعل الأداة (CheckBox١ نشطة
CheckBox١.Checked = True	جعل الأداة (CheckBox٢ نشطة
CheckBox١.Checked = False	جعل الأداة (CheckBox٣ غير نشطة

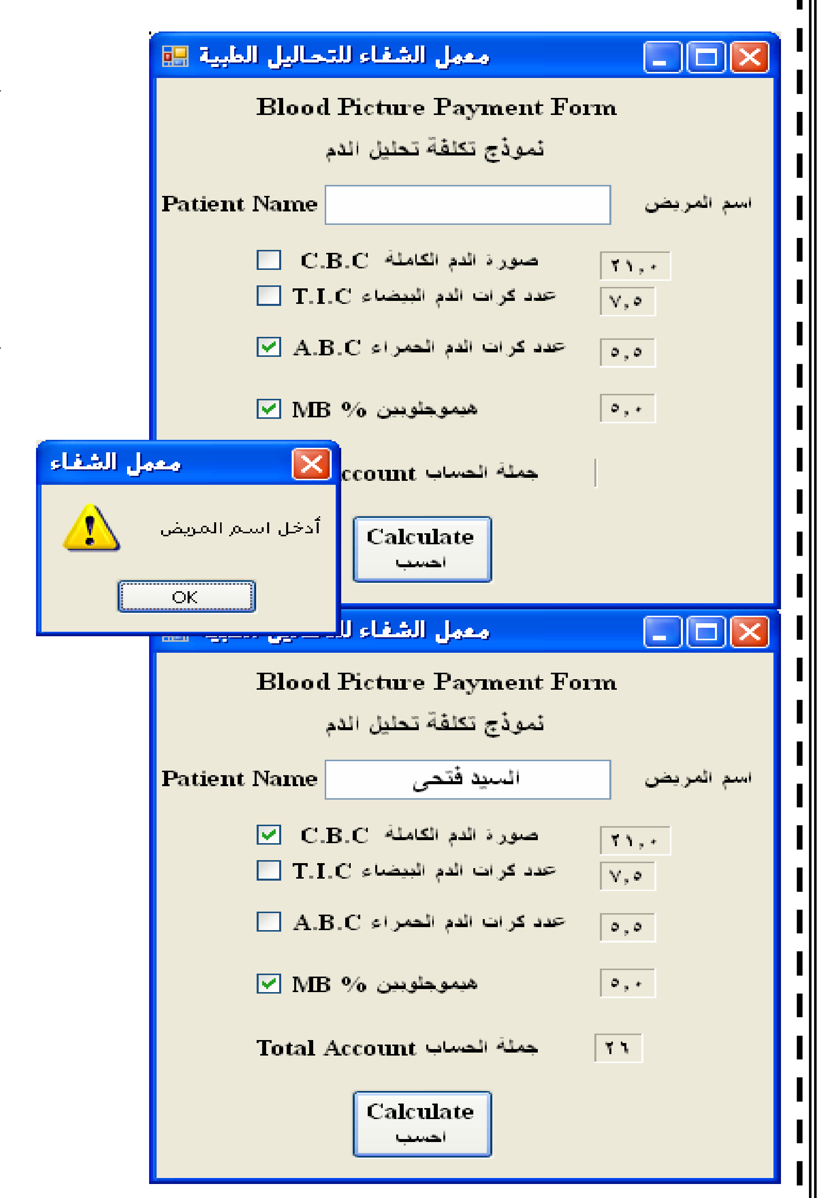
مثال : اكتب برنامج لمعمل تحاليل باستخدام أداة (CheckBox) وذلك لتحديد نوع التحليل ومعرفة إجمالى السعر . وحساب الفاتورة وفى حالة عدم إدخال اسم المريض يتم إظهار رسالة خطأ لعدم إدخال اسم المريض والجدول التالى يوضح سعر كل تحليل

سعر التحليل	اسم التحليل
٢١	صورة الدم الكاملة C.B.C
٧,٥	عدد كرات الدم البيضاء T.I.C
٥,٥	عدد كرات الدم الحمراء R.B.C
٥	هيموجلوبين HB

```
Public Class Form١

    Private Sub Button١_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button١.Click
        If Me.txtName.Text = "" Then
            MsgBox("أدخل اسم المريض", MsgBoxStyle.Exclamation, "معمل الشفاء")
        Else
            Dim total As Decimal
            If Me.ChkCBC.Checked = True Then
                total = total + ٢١
            End If

            If Me.ChkTIC.Checked = True Then
                total = total + ٧,٥
            End If
            If Me.ChkABC.Checked = True Then
                total = total + ٥,٥
            End If
            If Me.ChkMB.Checked = True Then
                total = total + ٥
            End If
            Me.lbltotal.Text = total.ToString
        End If
    End Sub
End Class
```



الأداة (GroupBox)

هى أداة تسمح للمستخدم أن يقسم عددا من الأداة **RadioButton** أو الأداة **CheckBox** إلى مجموعات ويتم الاختيار من هذه المجموعات حسب طبيعة كل أداة بداخل **GroupBox**.

والغرض الأساسى من استخدامها هو تنظيم وضع الأدوات على النموذج . حيث تظهر على شكل مستطيل أو مربع يحيط بمجموعة الأدوات

الخاصة به مثل(**RadioButton, Checkbox**) وتستخدم الخاصية **Text** لتحديد العلاقة التى تربط بين الأداة وما بداخلها من أدوات

كما يمكن التعامل مع الخصائص التالية والخاصة بهذه الأداة

الخاصية	الوظيفة
Name	تحديد اسم الأداة داخل نافذة البرمجة وعادة ما يبدأ الاسم بـ grp
Enabled	تحديد إمكانية جعل الأداة متاحة أم لا
Flat Style	تغيير شكل الأداة
Visible	جعل الأداة (مرئية / مخفية)

ليصبح الشكل العام للأداة كما يـلى :

مثال قم بتصميم نموذج لعرض البيانات الشخصية والذى يحتوى على البيانات التالية:

الاسم

السنة الدراسية: على أن يختار الطالب بين الصف (الأول – الثانى – الثالث)

النوع: على أن يختار الطالب بين (ذكر – أنثى)

الهوايات:

على أن يختار الطالب من بين الهوايات (كاراتيه – كرة قدم – موسيقى – رسم)



تصميم البرنامج:

١- أداة **Label** عنوان للاسم ٢- أداة **Textbox** لكتابة الاسم ٣- ٣ مجموعات الأولى للسنة الدراسية والثانية للنوع والثالثة للهوايات ٤- ٥ أدوات **RadioButton** للاختيار من (الأول – الثانى – الثالث) والاختيار من (ذكر- أنثى)

٥- ٤ أدوات **CheckBox** للاختيار من (كرة قدم – موسيقى – كاراتيه – رسم)

وبذلك يتم الإختيار بين أكثر من مجموعة حيث أن

المجموعة	النوع	الاختيار الأول	الاختيار الثانى	الاختيار الثالث	الاختيار الرابع	ملاحظات
الهواية	**Checkbox**	كاراتيه	رسم	كرة قدم	موسيقى	اختيار أى عدد
السنة الدراسية	**RadioButton**	الأول	الثانى	الثالث	-	اختيار واحد
النوع	**RadioButton**	ذكر	أنثى	-	-	اختيار واحد

الفصل الثالث أوامـــر التكرار ١- DO While … LOOP

هى أوامر تتحكم فى أعادة تنفيذ عمل(أمر) معين أو مجموعة من الأوامر حتى يتحقق شرط معين حيث كان يتم تنفيذ الأوامر فى أى برنامج من أعلى إلى أسفل باستثناء الأمرين (IF .. Then .. Else) , (Try .. Catch) حيث إنهما يقومان بتنفيذ بعض الأوامر مع تجاهل مجموعة من الأوامر الأخرى طبقاً لتحقيق شرط معين .

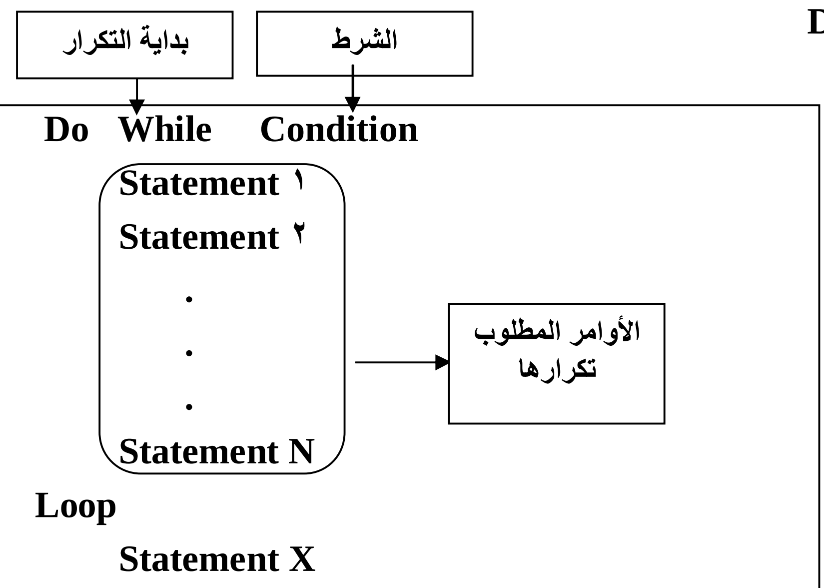
ولكن مع استخدام (Do .. Loop) سوف يتم تكرار مجموعة من الأوامر وفقا لتحقق شرط معين أو عدم تحققه.

أوامر التكرار متعددة منها:

يتم تنفيذ التكرار إذا كانت نتيجة الشرط True	١- Do While … Loop
يتم تنفيذ التكرار إذا كانت نتيجة الشرط False	٢- Do Until … Loop
يتم تنفيذ أمر معين ثم اختبار الشرط ويتم التكرار إذا كانت نتيجة الشرط True	٣- Do … Loop While
يتم تنفيذ أمر معين ثم اختبار الشرط ويتم التكرار إذا كانت نتيجة الشرط False	٤- Do …. Loop Until
يتم تنفيذ التكرار عدد معروف ومحدد من المرات	٥- For ………… Next

١- الأمر Do While --------------- Loop

القاعدة العامة



عند عمل مشروع باستخدام (Do While .. Loop) سوف يتم تقييم الشرط

(أ) إذا كان الشرط (False): سوف يتم تنفيذ مجموعة الأوامر التى تلى (Loop)

(ب) إذا كان الشرط (True)

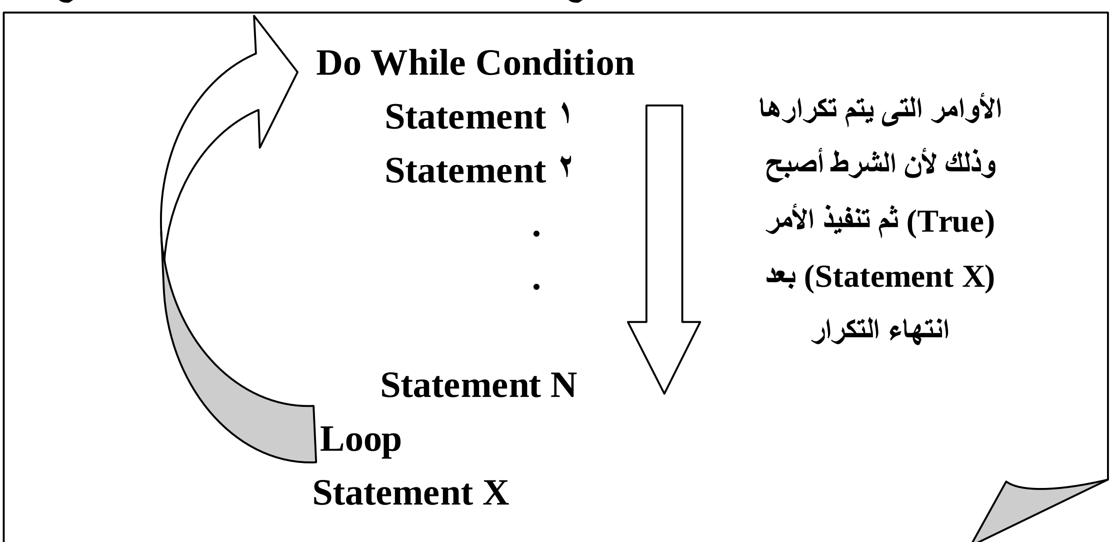
سوف يتم تنفيذ مجموعة الأوامر التى تلى الشرط (Condition) وحتى آخر أمر قبل (Loop) وعند وصول البرنامج إلى (Loop) فإنه يعود مرة أخرى إلى الأمر (Do While) ثم اختبار الشرط مرة أخرى فإذا كانت النتيجة True يتم التكرار أما إذا كانت False فتتوقف عملية التكرار ويتم تنفيذ الأوامر التى تلى كلمة Loop أما إذا لم توجد أوامر بعد Loop فينتهى عمل البرنامج

مثال اكتب برنامج لجمع الأعداد الفردية من واحد إلى ١٥ باستخدام جملة Do While----------- Loop

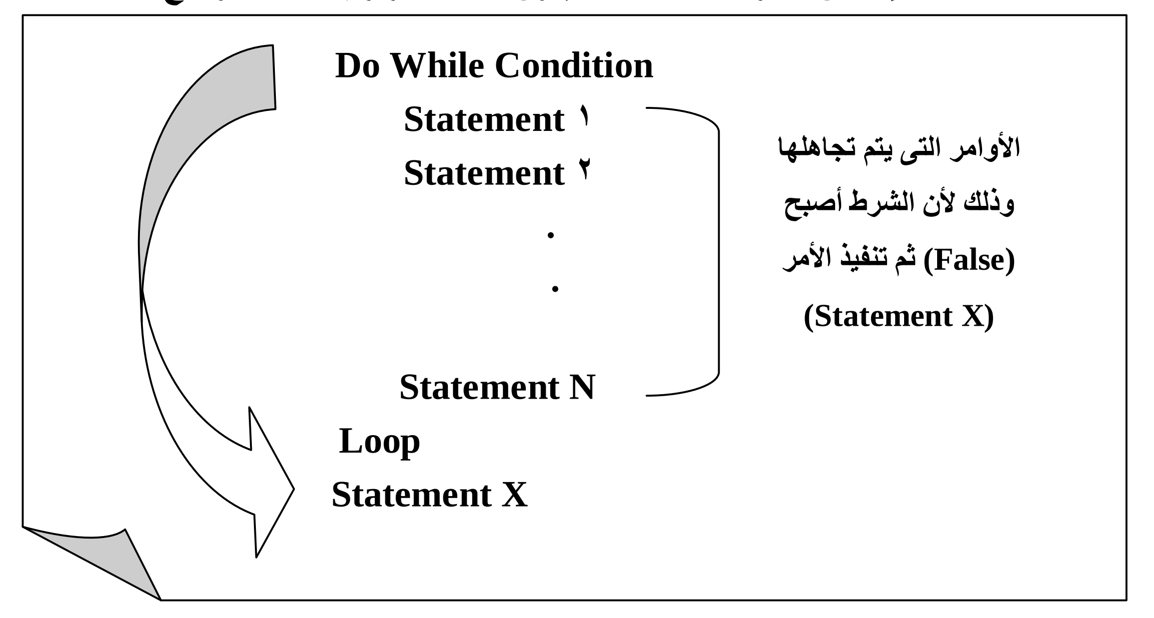
```
Dim Count, Sum As Integer
   Count=١
 Do While Count <= ١٥
    Sum=Sum + Count
    Count=Count + ٢
Loop
   Msgbox (Sum)
```

الناتج هو ٦٤

إذا كان الشرط صحيح True يكون شكل التكرار بالشكل الموضح

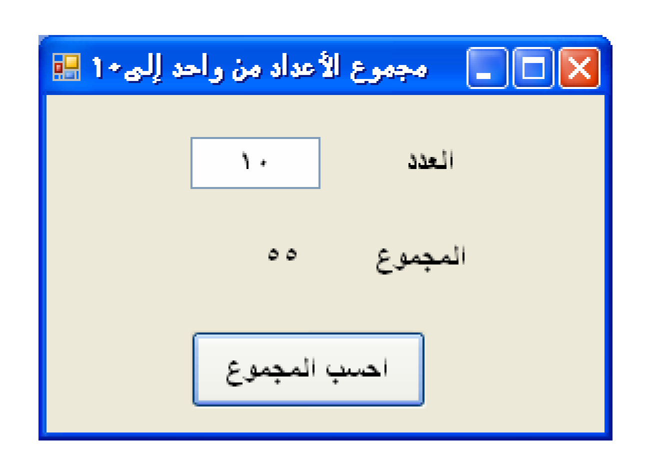


أما إذا كان الشرط خطأ False يكون شكل التكرار بالشكل الموضح



<u>مثال :</u> اكتب برنامج لجمع الأعداد (١،٢،٣ ....) حتى رقم معين باستخدام الأمر (Do While .. Loop)

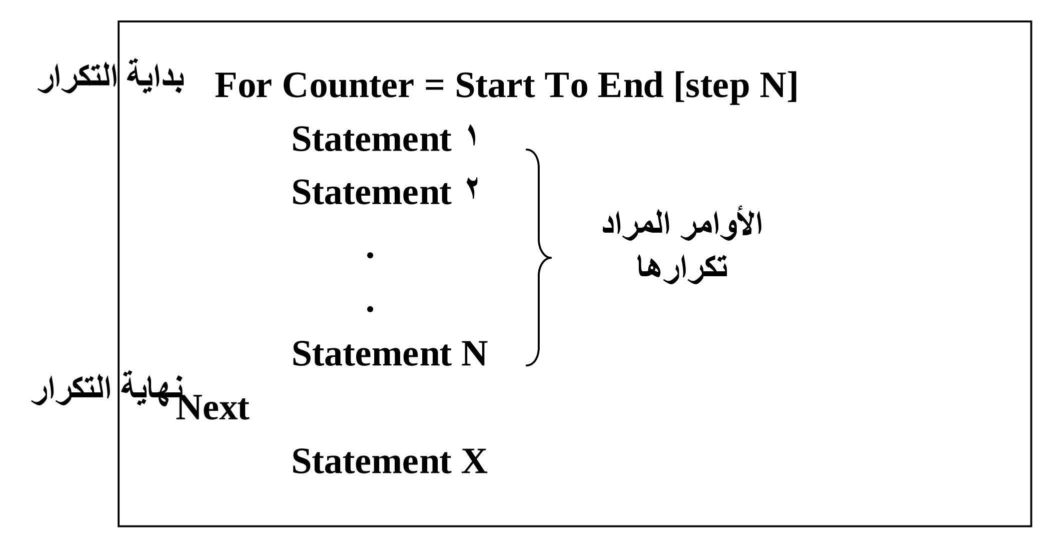
```
Dim num, x, sum As Integer
    num = Me.TextBox١.Text
    Do While x < num
        x = x + ١
        sum = sum + x
    Loop
    Me.Label٣.Text = (sum)
```



الأمـــر For … Next

يستخدم هذا الأمر فى تكرار تنفيذ أمر معين أو مجموعة من الأوامر عدد من المرات المحددة مسبقا

القاعدة العامة



يلاحظ عند صياغة الجملة السابقة الآتى:

١- (For) تشير لبداية التكرار ٢- (Next) تشير لنهاية التكرار

٣- (Counter) هو متغيرعداد يتم فيه تخزين عدد مرات التكرار بحيث يتم المقارنة بين (End, Start)

٤- (Start) تشير إلى القيمة المبدئية للمتغير(العداد) فى التكرار

٥- (End) تشير إلى القيمة النهائية للمتغير(العداد) فى التكرار

(يمكن أن تكون Start أكبر من End بشرط أن تكون قيمة Step بالسالب)

٦- (Step) تشير لمقدار الزيادة فى العداد وهى اختيارية وإذا لم توضع تكون زيادة العداد بمقدار واحد

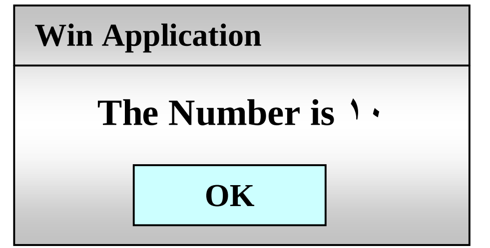
(أ) إذا كانت قيمة العداد (Counter) أكبر من قيمة (End) وينتهى التكرار ويتم تنفيذ الأمر (Statement X)

(ب) إذا كانت قيمة العداد (Counter) أقل من قيمة (End) يتم تنفيذ الأوامر التى تلى (For) حتى تصبح قيمة (Counter = End)

مثـــال: اكتب برنامج باستخدام الأمر (For.. Next) لعرض الأرقام من واحد إلى عشرة من خلال مربع الرسالة

Button١

```
Dim X As Integer
For X = ١ To ١٠
MsgBox (" The Number Is" & X)
Next
```



مثال: اكتب برنامج باستخدام (For.. Next) لإظهار الأرقام الفردية من (١) وحتى رقم معين

Button١

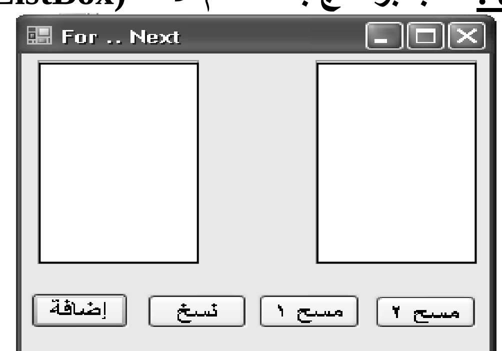
```
Dim  X  As  Integer
Dim  Y  As  Integer
Y = Integer. Parse (TextBox١.Text)
For  X = ١  To  Y  Step٢
MsgBox (" Number Is     " & X)
Next
```

الأداة ListBox

هذه الأداة واحدة من الأدوات الخاصة بالتكرار . وتستخدم فى إنشاء وعرض قائمة من العناصر للاختيار منها. يمكن أن تظهر بها أشرطة تمرير إذا كانت مجموعة العناصر لا تتناسب مع حجم الأداة ومن هنا يوجد بعض الخصائص التى تتعامل مع هذه الأداة ومنها

الخاصيـــة	الأداة
الاسم الأصلى للأداة داخل نافذة البرمجة	Name
تحديد ما إذا كانت الأداة بها شريط تمرير	HorizontalScrollbar
معرفة العناصر الموجودة داخل الأداة	Items
تحديد العنصر الواقع عليه الإختيار فى الأداة	Selected Items
ترتيب العناصر داخل الأداة ترتيباً أبجدياً	Sorted

مثال : اكتب برنامج باستخدام الأداة (ListBox) يقوم بإجراء عمليتى النسخ والمسح بين أداتى (ListBox)





Button١(Add)

ListBox١.Items.Add ("Saturday")
ListBox١.Items.Add ("Sunday")
ListBox١.Items.Add ("Monday")
ListBox١.Items.Add ("Tuesday")
ListBox١.Items.Add ("Wednesday")
ListBox١.Items.Add ("Thursday")
ListBox١.Items.Add ("Friday")

Button٢ (Copy)

ListBox٢.Items.Add (ListBox١.Selecteditem)

Button٣(Clear)

ListBox١.Items.Clear

Button٤(Clear Choose)

ListBox٢.Items.Clear

١- عند الضغط على مفتاح (Button١) سوف يتم عرض أيام الأسبوع
٢- عند الضغط على مفتاح (Button٢) سوف يتم نسخ العنصر المحدد من(ListBox١) إلى (ListBox٢)
٣- عند الضغط على مفتاح (Button٣) سوف يتم مسح محتوى (ListBox١)
٤- عند الضغط على مفتاح (Button٤) سوف يتم مسح محتوى (ListBox٢)

ويلاحظ أن الخاصية Items للأداة ListBox كائن له خصائص ووسائل مثل

العمل	الطريقة
إضافة ما بين القوسين كعنصر أخير فى القائمة	Add(text)
معرفة عدد العناصر	Count
إضافة العنصر Text بعد عنصر له رقم محدد Index	Insert(index, Text)
حذف عنصر فى القائمة قيمته هى Text	Remove(text)
إزالة العنصر المحدد مكانه بالرقم Index	RemoveAt(Index)
إفراغ الأداة من العناصر	Clear

الفصل الرابع الميقاتى TIMER

الكائن Timer هو ساعة توقيت غير مرئية تتيح لك التعامل مع ساعة النظام (System Clock) من داخل البرامج التى يتم تنفيذها من خلال برنامج VB.Net

وتأتى الحاجة إلى الأداة Timer لتنفيذ أمر أو مجموعة من الأوامر بطريقة تلقائية كل فترة زمنية بدون تدخل من المستخدم ، وذلك بدلا من النقر على أداة مثل الزر Button فى كل مرة نحتاج لتنفيذ هذه الأوامر.

ويتوقف التعامل مع الأداة Timer على مجموعة من الخصائص والأحداث هى:

الخاصية Interval	هذه الخاصية تحدد الفترة الزمنية اللازمة لتكرار تنفيذ الأوامر ، وتقاس بالمللى ثانية وتتراوح قيمتها بين صفر و ٦٥٥٣٥ حيث أن كل ثانية تعادل ١٠٠٠مللى ثانية
الخاصية Enabled	تسمى خاصية التمكين أو الإتاحة وتستخدم فى تشغيل الأداة Timer والوضع الافتراضى لها False ولكى يتم تنفيذ أوامر الـ Timer لابد من تغييرها إلى True ويمكن تغيير هذه الخاصية أثناء تصميم البرنامج وأيضا فى مرحلة التنفيذ أو التشغيل
الحدث Tick	هو الحدث المصاحب للأداة Timer وينشط فى كل مرة مرة مع الخاصية Interval والخاصية Enabled ويكتب فيه الأوامر(والتى هى عبارة عن برنامج صغير "Procedure") والمطلوب تنفيذها

مثال : اكتب برنامج باستخدام أداة (Timer) لتصميم ساعة رقمية تعلن عن الوقت والتاريخ الحاليين ومفتاحى (تشغيل وإيقاف)



Timer١

Label٢.Text=Date. Now

Button١ (إيقـــاف)

Timer١.Enabled = False

Button٢(تشغيل)

Timer١.Enabled = True

ولكن قبل البدء فى كتابة البرنامج لابد من ضبط خصائص الأداة (Timer) وهما (Enable , Interval)

Enabled=true

Interval=١٠٠٠

التكوين DateTime

بعض خصائص التاريخ والوقت

عند الإعلان عن متغير من النوع Date يمكن استعمال التكوين (DateTime)ولهذا التكوين العديد من الخصائص والطرق Methods

هناك بعض خصائص التاريخ والوقت التي يمكن استخدامها لمعرفة التاريخ والوقت الحالي الموجود بجهاز الكمبيوتر. كما تساعدك أيضاً في معرفة أجزاء التاريخ وأجزاء الوقت الجدول التالي يوضح هذه الخصائص .

اسم الخاصية	الغرض منهـــا
Now	معرفة التاريخ والوقت الحاليين الموجودين بجهاز الكمبيوتر
Today	معرفة التاريخ الحالي الموجود بجهاز الكمبيوتر
TimeOfDay	معرفة الوقت الحالي الموجود بجهاز الكمبيوتر
Year	معرفة السنة من تاريخ معين
Month	معرفة الشهر من تاريخ معين
Day	معرفة اليوم من تاريخ معين
DayOfWeek	معرفة يوم الأسبوع من تاريخ معين
Hour	معرفة الساعة الموجودة في وقت معين
Minute	معرفة الدقائق الموجودة في وقت معين

ولتوضيح هذه الخصائص ، نستعرض الآن بعض الأمثلة كالآتى :

مثال (١)

الناتج والتوضيح		التكوين
الإعلان عن متغير من النوع تاريخ		Dim X As Date
١٢:٠٠:٠٠ AM		Msgbox(x)
تحديد قيمة للمتغير وتاريخ ووقت محدد		X = #٤/٢٠/٢٠١٠ ٨:٤٥:١٣ AM#
معرفة الثوانى فى الوقت المحدد	١٣	١. MsgBox(x.Second)
معرفة الدقائق فى الوقت المحدد	٤٥	٢. MsgBox(x.Minute)
معرفة الساعات فى الوقت المحدد	٨	٣. MsgBox(x.Hour)
معرفة اليوم فى التاريخ المحدد	٢٠	٤. MsgBox(x.Day)
معرفة الشهر فى التاريخ المحدد	٤	٥. MsgBox(x.Month)
معرفة اليوم فى التاريخ المحدد	٢٠١٠	٦. MsgBox(x.Year)
معرفة اليوم فى الأسبوع للتاريخ المحدد	٤	٧. MsgBox(x.DayOfWeek)
معرفة اليوم فى الأسبوع للتاريخ المحدد كنص Tuesday		٨. MsgBox(x.DayOfWeek.ToString)
معرفة رقم اليوم فى السنة لالتاريخ المحدد ١١٠		٩. MsgBox(x.DayOfYear)

يلاحظ أن القيمة المبدأية لمتغير التاريخ إذا لم تخصص له قيمة مبدأية تكون ١٢:٠٠:٠٠ AM أى بداية يوم جديد

فى المثال رقم٧ (Msgbox(x.DayOfWeek كان الناتج عرض اليوم فى الأسبوع كرقم أى رقم اليوم فى الأسبوع

فى المثال رقم٨ (Msgbox(x.DayOfWeek.ToString كان الناتج عرض اليوم فى الأسبوع كنص

أمثلة على التعامل مع

التكوين	الناتج والتوضيح
MsgBox(Now)	عرض الوقت والتاريخ المسجل فى ذاكرة الكمبيوتر
MsgBox(Today)	عرض التاريخ الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(TimeOfday)	عرض الوقت الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Now.Second)	عرض الثوانى للوقت الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Now.Minute)	عرض الدقائق للوقت الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Now.Hour)	عرض الساعات للوقت الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Today.Day)	عرض اليوم للتاريخ الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Today.DayOfWeek)	عرض اليوم فى الأسبوع للتاريخ الحالى المسجل فى ذاكرة الكمبيوتر
MsgBox(Today.DayOfWeek. ToString)	عرض اليوم فى الأسبوع للتاريخ الحالى المسجل فى ذاكرة الكمبيوتر كنص
MsgBox(Today.DayOfYear)	عرض اليوم فى السنة للتاريخ الحالى المسجل فى ذاكرة الكمبيوتر

بعض الطرق الأخرى المستخدمة لتعديل الوقت أو التاريخ

ملحوظة لا يمكن تعديل أو تغيير المتغير من النوع (Date) باستخدام عمليات الجمع والطرح العادية.

Dim Y as date

Method الطريقة	Purposeالغرض	Exampleمثال	Result النتيجة
AddDays	إنشاء تاريخ جديد متقدم أو متأخر عدد من الأيام	Y.AddDays(١) Y.AddDays(-٢)	تقديم يوم واحد تأخير يومان
AddHours	إنشاء تاريخ جديد متقدم أو متأخر عدد من الساعات	Y.AddHours(٨) Y.AddHours(-٥)	تقديم ٨ ساعات تأخير ١٠ساعات
AddMinutes	إنشاء تاريخ جديد متقدم أو متأخر عدد من الدقائق	Y.AddMinutes(٢٠) Y.AddMinutes(-١٥)	تقديم ٢٠ دقيقة تأخير ١٥ دقيقة

مثال: إذا علمت **Dim Y as Date=#٤/١٨/٢٠١٠ ٦:٣٥:١٢ AM#**

قم بعرض ناتج قيمة Y وفقا للتعديلات فى الجدول السابق (على أن يتم التعامل مع كل قيمة على حدة)

Y=Y.AddDays(١) **Msgbox(y.day)** **الناتج هو ١٩**

Y=Y.AddDays(-٢) **Msgbox(y.day)** **الناتج هو ١٦**

Y=Y.AddHours(٨) **Msgbox(y.Hour)** **الناتج هو ١٤**

Y=Y.AddHours(-٥) **Msgbox(y.Hour)** **الناتج هو ١**

Y=Y.AddMinutes(٢٠) **Msgbox(y. Minute)** **الناتج هو ٥٥**

Y=Y.AddMinutes(-١٥) **Msgbox(y. Minute)** **الناتج هو ٢٠**

أسئلة وتدريبات

السؤال الأول ضع علامة (✓) أمام العبارة الصحيحة وعلامة (X) أمام العبارة الخاطئة

١- يمكن أن يبدأ اسم المتغير برقم.
٢- إذا كتبت axd=٣ فأن هذا يعتبر ثابتا.
٣- يسمح بوضع مسافة فى اسم المتغير.
٤- عند حفظ البيانات فى الذاكرة فى فيجوال بيزيك ، تحفظ البيانات الرقمية والبيانات الحرفية فى صورة أرقام.
٥- الكلمة Double تصلح أن تكون اسما لثابت.
٦- يستخدم الأمر If للتحكم فى ترتيب تنفيذ أوامر البرنامج.
٧- يستعمل الأمر Do While .... Loop فى تكرار أمر معين ، أو مجموعة من الأوامر.
٨- لا يصلح الاسم Dim لتعريف اسم متغير، لأنه من الكلمات المحجوزة .Reserved Words
٩- تستخدم مجموعة أوامر Try / Catch لاكتشاف الأخطاء التى تحدث أثناء تشغيل البرنامج والتعامل معها.
١٠- تستخدم الكلمة Dim للإعلان عن الثوابت.
١١- الأداة CheckBox تتيح للمستخدم مجموعة من البدائل لاختيار واحدة منها فقط.
١٢- يبدأ تنفيذ العمليات الحسابية من اليسار لليمين ثم من الأعلى فى الأسبقية إلى الأقل فى الأسبقية.
١٣- يمكن استخدام الأمر Dim للإعلان عن أكثر من متغير فى نفس السطر.
١٤- أول أولويات تنفيذ العمليات الحسابية هى الجمع والطرح.
١٥يمكن أن يختار المستخدم فى نفس الوقت ، أى عدد يريده من الاختيارات التى تعرضها مربعات الاختيار Checkbox
١٦- من الآمن تحويل البيانات من النوع Double إلى أى نوع آخر باستخدام طريقة التحويل المباشر.
١٧- يستخدم الأمر (If… ElseIf… Else) إذا كان هناك بديلان محتملان لمسار البرنامج.
١٨- تستخدم الدالة Val( ) لمنع المدخلات الخاطئة من إيقاف تنفيذ البرنامج.
١٩- يمكن جمع أو طرح ثوانى ودقائق وساعات من المتغيرات من النوع Date.
٢٠- تستخدم المعاملات المنطقية لإجراء العمليات الحسابية على القيم العددية.
٢١- المتغير الثابت Static Variable يشبه الثابت Constant فى أن قيمته لا تتغير طوال تشغيل البرنامج.
٢٢- يمكن استخدام معاملات المقارنة < و > و = و < > لمقارنة الحروف والنصوص.
٢٣- ناتج العملية الحسابية هو قيمة عددية أو حرفية يمكن وضعها فى متغير حرفى آخر.
٢٤- تقبل المتغيرات الحرفيةChar والنصية String فى الفيجوال بيزيك وضع قيم رقمية كمحتوى لها.
٢٥- عندما تكون للخاصية Interval القيمة ٢٠٠٠ فإن هذا معناه أن حدث التنبيه (التكة) الخاص بالمنبه سينطلق كل ٢٠ ثانية.
٢٦- يمكن استخدام العلامة = فى فيجوال بيزيك كأمر تخصيص وكمعامل مقارنة منطقى.
٢٧- بعد تنفيذ أمر التخصيص Assignment تتغير قيمة المتغير الموجود على يمين علامة =.
٢٨ يستخدم الرمز & للربط بين شرطين.
٢٩- عند حفظ ١٢٣ و "١٢٣" فى الذاكرة فإن كلا منهما يحتاج إلى خلية واحدة فقط .(١ Byte)
٣٠- عند تداخل الأقواس فى العمليات الحسابية ، يتم حساب قيمة الأقواس الداخلية اولا ثم الأقواس الخارجية.
٣١- الشرط التالى صحيح :True “Ali” > “Ahmed”

إجابة السؤال الأول:

رقم السؤال	الإجابة	التوضيح					
١	خطأ	لابد أن يبدأ اسم المتغير بحرف من الحروف الإنجليزية ويمكن أن يحتوى على رقم.					
٢	خطأ	هذه الجملة أمر تخصيص أما الإعلان عن ثابت فيكون Const axd as Integer=٣					
٣	خطأ	لا يسمح بوضع مسافة ولا نقطة فى اسم المتغير.					
٤	صح	لأن جميع أنواع البيانات يتم تحويلها إلى أرقام كى يستطيع الحاسب التعامل معها.					
٥	خطأ	لأن الكلمة Double من الكلمات المحجوزة ولا تصلح لأن تكون اسم متغير أو ثابت.					
٦	صح	٧	صح	٨	صح	٩	صح
١٠	خطأ	تستخدم الكلمة Dim للإعلان عن المتغيرات بينما الكلمة Const للإعلان عن الثوابت.					
١١	خطأ	تتيح للمستخدم اختيار أى عدد منها فى نفس الوقت بينما الأداة RadioButton اختيار واحد فقط					
١٢	خطأ	يبدأ تنفيذ العمليات الحسابية من الأعلى فى الأسبقية إلى الأقل فى الأسبقية ثم من اليسار لليمين					
١٣	صح						
١٤	خطأ	أول أولويات تنفيذ العمليات الحسابية هو الأقواس بينما الجمع والطرح آخر أولويات التنفيذ.					
١٥	صح						
١٦	خطأ	التحويل المباشر غير آمن نظرا لأن النوع Double قد يكون العدد المخزن فيه أكبر سعة					
١٧	خطأ	يستخدم هذا الأمر إذا كان هناك أكثر من بديلان محتملان لمسار البرنامج.					
١٨	صح	لأنها تحول النص الفارغ أو الغير رقمى إلى صفر والرقمى تحوله إلى عدد صحيح.					
١٩	صح						
٢٠	خطأ	تستخدم المعاملات المنطقية لعمل الشروط بينما تستخدم المعاملات الحسابية لإجراء العمليات الحسابية					
٢١	خطأ	هو متغير يحتفظ بقيمته بعد انتهاء الإجراء ولكن يمكن تغييرها داخل الإجراء نفسه.					
٢٢	صح						
٢٣	خطأ	ناتج العملية الحسابية هو قيمة عددية يمكن وضعها فى متغير عددى آخر.					
٢٤	خطأ	خطأ لأن المتغير الحرفى Char لا يقبل قيمة رقمية بينما الـ String يقبل أرقام وتعامل كحروف					
٢٥	خطأ	خطأ لأنه سينطلق كل ٢٠٠٠/١٠٠٠ = ٢ أى كل ثانيتين					
٢٦	صح						
٢٧	خطأ	بعد تنفيذ أمر التخصيص تتغير قيمة المتغير على يسار علامة= بينما المتغيرات على اليمين لا تتغير					
٢٨	خطأ	يسمى المعامل & معامل الربط لربط أو تجميع سلسلتين حرفيتين.					
٢٩	خطأ	يحتاج الرقم١٢٣ إلى Byte واحدة بينما يحتاج النص "١٢٣" إلى ثلاث خلايا.					
٣٠	صح						
٣١	صح	لأن l أكبر من h					

س٢: اكتب الخطوات اللازمة لحل المعادلتين التاليتين:

١) X= ٣٨٢*٢+٤*٣

٢) Y=٥+٦*٢/ (٣+١)

الحـــــــــــــل

١)

X=٩*٢+٤*٣ — X=١٨+١٢ — X=٣٠

٢)

Y=٥+٦*٢/٤ — Y=٥+١٢/٤ — Y=٥+٣ — Y=٨

س٣:اختر الإجابة الصحيحة لما يلى:

١- يستخدم الكائن ----------- لجعل بعض الأحداث تقع بعد فترة زمنية معينة وبدون تدخل من المستخدم.
أ- Color ب- Form ج- Timer د- DateTime هـ- Date

٢- -------------- تكون سابقة التعريف فى VB.Net ويستطيع المبرمج استخدامها مباشرة.
أ- المتغيرات ب- الثوابت المنطقية ج- الثوابت المسماة د- الثوابت الجوهرية هـ- السلاسل الحرفية

٣- الخاصية Interval للميقاتى Timer تحدد معدل وقوع الحدث بالوحدة الزمنية -----------
أ- مللى ثانية ب- ثانية ج- دقيقة د- ساعة هـ- لحظة

٤- يطلق على المتغير الذى يتحكم فى عدد مرات تكرار تنفيذ الأوامر اسم ---------------
أ- Loop ب- Counter ج- A د- Control هـ- Timer

٥- إذا كان تعريف المتغير X هو: Dim X As Integer=٥,٢
فإن قيمة X هى: أ- ٢ ب- ٥ ج- ٥,٢ د- ٢, هـ- ٦

٦- لتحويل القيمة الحرفية إلى قيمة رقمية نستخدم:
أ- Convert.ToInteger ب- Val ( ) ج- Integer. Parse د- ToString هـ- كل ما سبق ماعدا د

٧- يمكن تخزين الأعداد التى تحتوى على كسور عشرية فى متغيرات من النوع:
أ- Single ب- Double ج- Decimal د- Long هـ- كل ما سبق ماعدا د

٨- ناتج العملية الحسابية ١-٢^٨٥*٢+٣ هـــــو:
أ- ٦٢٤ ب- ١٢٤ ج- ٢٥ د- ٥٢ هـ- ١٠٢

٩- إذا أردنا تخزين القيمة ٧٢٣٤٥٣,٢ فمن الأفضل وضعها فى متغير من النوع:
أ- Single ب- Double ج- Decimal د- Long هـ- Integer

١٠- تستخدم Try ...... Catch فى:
أ- تجاهل الأخطاء ب- اكتشاف الأخطاء ج- إنتاج الأخطاء د- تصحيح الأخطاء هـ- ج و د معا.

١١- العملية التى يتم بها التحويل من نوع بيانات رقمى ضيق Narrow إلى نوع أكثر اتساعا Wider يطلق عليها:
أ- تحويل البيانات ب- التحويل التلقائى ج- تغيير البيانات د- التحويل غير التلقائى هـ- توسيع البيانات

١٢- لإجراء التحويل الصريح (غير التلقائى) Explicit نستخدم:
أ- Convert ب- Val( ) ج- ToString د- Parse هـ- كل ما سبق

١٣- الرمز "" فى VB.Net يرمز أو يشير إلى:
أ- القيمة صفر ب- سلسلة حرفية فارغة ج- الرمز الخالى د- المسافة هـ- علامتى التنصيص

١٤- يمكن تعريف متغيرين من النوع Integer وثلاث متغيرات من النوع String كالتالى:

أ- Dim N١, N٢ As Integer , S١, S٢,S٣ As String

ب- Dim N١, N٢ As Integer
Dim S١, S٢,S٣ As String

ج- Dim n١ As Integer
Dim N٢ As Integer
Dim S١ As String
Dim S٢ As String
Dim S٣ As String

د- Dim N١, N٢ As Integer : S١, S٢,S٣ As String

هـ- كل ما سبق صحيح

١٥- عند وضع قيمة فى عنوان (متغير) معين فى ذاكرة الحاسب فإن هذه القيمة:
أ- تضاف للقيمة السابقة لهذا العنوان ب- تحل محل القيمة السابقة لهذا العنوان
ج- يتم ضربها فى القيمة السابقة فى هذا العنوان د- توضع فى عنوان الذاكرة التالى لأن العنوان الحالى مشغول
هـ- يتم رفضها لأن هذا العنوان مشغول

١٦- عند قراءة قيمة فى عنوان (متغير/ ثابت) معين فى ذاكرة الحاسب:
أ- يتم محو هذه القيمة من هذا العنوان   ب- يتم نقل هذه القيمة من هذا العنوان
ج- لا تتأثر هذه القيمة وتظل كما هى د- يوضع بدلا من هذه القيمة نص فارغ Empty String
هـ- يوضع صفر بدلا من هذه القيمة فى الذاكرة.
١٧- تحسب قيمة التعبير الموجود على يمين علامة = فى أمر التخصيص:
أ- قبل إجراء عملية التخصيص   ب- أثناء إجراء عملية التخصيص   ج- بعد إجراء عملية التخصيص
د- بعد حساب قيمة المتغير على يسار علامة =   هـ- لا يتم احتساب هذه القيمة مطلقا.
١٨- لإجراء عملية القسمة الصحيحة نستخدم:
أ- Div   ب- Mod   ج- /   د- \   هـ- ÷
١٩- تحسب قيمة العمليات الحسابية :
أ- من اليسار لليمين   ب- من اليمين لليسار   ج- من المنتصف
د- من الأعلى أولوية إلى الأقل أولوية.   هـ- من الأقل أولوية إلى الأعلى أولوية.
٢٠- يقوم الأمر If .... Then بعملية تسمى:
أ- Single Selection   ب- Double Selection   ج- Multi Selection
د- No Selection   هـ- Repetition
٢١- استخدام المعامل ---------- يعطى False إذا كان ما على يساره أقل مما على يمينه:
أ- >   ب- <   ج- >=   د- =   هـ- كل ما سبق ما عدا أ
٢٢- الخاصية ---------- تضع نصا على أداة الاختيار CheckBox :
أ- Label   ب- String   ج- Text   د- Name   هـ- Checked
٢٣- الأمر الذى يتم تنفيذه حتى يصبح الشرط خاطئا هو:
أ- Do While ------ Loop   ب- If ------- Then   ج- For ------- Next
د- Try -------- Catch   هـ- If ------ Then ------ Else
٢٤- الوسيلة Method التى تمحو كل عناصر القائمة ListBox١ هى:
أ- ListBox١.Items.Clear   ب- ListBox١.Clear   ج- ListBox١.Items.RemoveAll
د- ListBox١.RemoveAll   هـ- ListBox١.Items.Remove
٢٥- الوسيلة Method التى تضيف عنصرا إلى القائمة ListBox١ هى:
أ- ListBox١.Items.Add   ب- ListBox١.Items.Insert   ج- ListBox١.Items.Append
د- ListBox١.Items.Include   هـ- أ و ب معا
٢٦- الدالة Now تخبرنا بمعلومة تخص جهاز الحاسب وهى:
أ- الوقت الحالى   ب- التاريخ الحالى   ج- التاريخ والوقت الحالى   د- الوقت المنقضى منذ تشغيل الكمبيوتر
٢٧- يمكن طرح يوم من التاريخ الموجود فى المتغير D باستخدام الأمر:
أ- D=D.AddHours (-٢٤)   ب- D=D.AddDays (-١)   ج- D=D.AddMinutes (-٢٤*٦٠)
د- D=D.AddSecond (-٢٤*٦٠*٦٠)   هـ- كل ما سبق.
٢٨- تستطيع الخلية الواحدة فى الذاكرة تخزين:
أ- Byte   ب- Short   ج- Integer   د- Long   هـ- Double
٢٩- نوع البيان يحدد:
أ- نوع البيانات التى يتم تخزينها فيه ب- نوع العمليات التى سيتم إجراؤها عليه ج- حجم الذاكرة التى يحتاجها.
د- أ و ب   هـ- أ و ب و ج
٣٠- يوضع الرمز ---------- فى بداية سطر التعليق:
أ- /   ب- &   ج- :   د- '   هـ- ;

إجابة السؤال الثالث:

سؤال رقم	الإجابة الصحيحة	سؤال رقم	الإجابة الصحيحة
١	ج- Timer	١٦	ج- لا تتأثر هذه القيمة وتظل كما هى
٢	د- الثوابت الجوهرية	١٧	أ- قبل إجراء عملية التخصيص
٣	أ- مللى ثانية	١٨	د- \
٤	ب- Counter	١٩	د- من الأعلى أولوية إلى الأقل أولوية.
٥	ب- ٥	٢٠	أ- Single Selection
٦	هـ- كل ما سبق ماعدا د	٢١	هـ- كل ما سبق ما عدا أ
٧	هـ- كل ما سبق ماعدا د	٢٢	ج- Text
٨	د- ٥٢	٢٣	أ- Do While ------ Loop
٩	أ- Single	٢٤	أ- ListBox١.Items.Clear
١٠	ب- اكتشاف الأخطاء	٢٥	هـ- أ و ب معا
١١	ب- التحويل التلقائى	٢٦	ج- التاريخ والوقت الحالى
١٢	هـ- كل ما سبق	٢٧	هـ- كل ما سبق
١٣	ب- سلسلة حرفية فارغة	٢٨	أ- Byte
١٤	هـ- كل ما سبق صحيح	٢٩	هـ- أ و ب و ج
١٥	ب- تحل محل القيمة السابقة لهذا العنوان	٣٠	د- '

السؤال الرابع أكمل العبارات التالية:

(١) ٤٨ Mod ٥ = ------ (٢) ٢*(٣+٢)^٢=

٣- يستخدم الأمر ---------------------- للتعامل مع الأخطاء.

٤- يستخدم الأمر ---------------------- لزيادة قيمة العداد وتكرار تنفيذ الكود (الأوامر)

٥- تنتهى جملة If ----- Then ----- Else بـ ----------------

٦- لتحويل قيمة رقمية لسلسلة حرفية نستخدم ----------------------

٧- إذا كانت القيم المخزنة تتغير أثناء تشغيل البرنامج فإن أسماء خلايا الذاكرة التى تحفظ هذه القيم تسمى ----------

٨- ------------ لها قيمة ثابتة لا تتغير أثناء تشغيل البرنامج.

١- ٣	٢- ٥٠	٣- Try --- Catch	٤- For ----- Next
٥- End If	٦- ToString	٧- المتغيرات	٨- الثوابت

مع أطيب وأرق الأمنيات بالنجاح والتفوق